

**BIDIRECTIONAL GRU DENGAN ATTENTION MECHANISM
PADA ANALISIS SENTIMEN PLN MOBILE**

THESIS

**Oleh:
MOH. AINUR ROHMAN
NIM. 210605210005**



**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PENGAJUAN

**BIDIRECTIONAL GRU DENGAN ATTENTION MECHANISM
PADA ANALISIS SENTIMEN PLN MOBILE**

THESIS

**Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Magister Komputer (M.Kom)**

**Oleh:
MOH. AINUR ROHMAN
NIM. 210605210005**

**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

DAFTAR ISI

HALAMAN PENGAJUAN	i
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
SURAT PERNYATAAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
ملخص.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah.....	5
1.6 Sistematika Penulisan	5
BAB II STUDI PUSTAKA	7
2.1 Analisis Sentimen	7

2.1.1 Sentimen Klasifikasi (<i>Task</i>)	8
2.1.2 <i>Deep Learning (Approaches)</i>	8
2.1.3 <i>Document-Based (Level of Analysis)</i>	9
2.2 <i>Bidirectional Gated Recurrent Unit (BiGRU)</i>	10
2.3 <i>Attention Mechanism (Transformer)</i>	12
2.4 Novelty Penelitian	14
2.5 Kerangka Teori	19
BAB III METODOLOGI PENELITIAN	21
3.1 Desain Penelitian	21
3.2 Pengumpulan Data (<i>Data Gathering</i>)	22
3.3 Pengolahan Data Awal (<i>Data Pre-processing</i>)	23
3.4 Model yang Diusulkan (<i>Proposed Method</i>)	27
3.5 Eksperimen dan Pengujian Model (<i>Model Test and Experiment</i>)	31
3.6 Evaluasi dan Validasi Hasil (<i>Result Evaluation and Validation</i>)	33
3.7 Desain Sistem	34
BAB IV HASIL DAN PEMBAHASAN	36
4.1 <i>Experiment Environment</i>	36
4.2 <i>Dataset</i>	36
4.3 Indikator Evaluasi	37
4.4 <i>Hyperparameter Setting</i>	38
4.4.1 <i>Dropout</i>	38
4.4.2 <i>Optimizer</i>	41
4.4.3 <i>Batch Size</i>	43

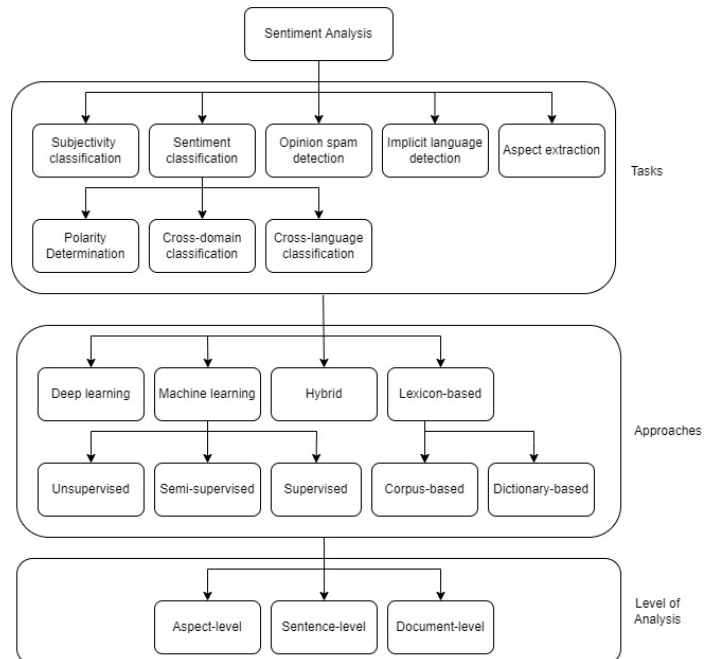
4.4.4 <i>Units</i>	46
4.4.5 <i>Learning Rate</i>	48
4.5 Komparatif Eksperimen	50
4.5.1 Perbandingan menggunakan dan tanpa <i>global attention</i>	51
4.5.2 Perbandingan dengan metode analisis sentimen yang lain	52
4.6 Memahami Informasi Penting dalam Al-Quran.....	55
BAB V PENUTUP	58
5.1 Kesimpulan	58
5.2 Saran.....	59
DAFTAR PUSTAKA	60

BAB II

STUDI PUSTAKA

2.1 Analisis Sentimen

Analisis sentimen adalah ilmu yang digunakan untuk menganalisis pendapat, sentimen, sikap dan emosi orang terhadap entitas seperti layanan, produk, organisasi, peristiwa dan atribut-atributnya (Bing Liu, 2012). Analisis sentimen memungkinkan untuk mengetahui sentimen masyarakat tentang entitas tertentu untuk menciptakan sebuah pengetahuan yang dapat ditindaklanjuti. Pengetahuan ini digunakan untuk memahami, menjelaskan, dan memprediksi fenomena sosial (Pozzi et al., 2017). Pada penelitian (Hemmatian & Sohrabi, 2019; Kumar & Jaiswal, 2020; Mite-Baidal et al., 2018; Pozzi et al., 2017) menggambarkan terkait konsep-konsep sentimen analisis seperti pada gambar 2.1.



Gambar 2.1 Konsep analisis sentimen

2.1.1 Sentimen Klasifikasi (*Task*)

Sentimen klasifikasi adalah salah satu *task* dari analisis sentimen yang bertugas untuk mengidentifikasi polaritas sentimen di setiap dokumen teks. Umumnya, polaritas diklasifikasikan sebagai positif dan negatif. Beberapa penelitian menganggap kelas ketiga adalah sentimen netral. *Task* ini adalah salah satu yang paling banyak dikenal dan diteliti dalam analisis sentimen (Wang et al., 2014). Beberapa penelitian terkait sentimen klasifikasi:

Penelitian yang dilakukan (Duan et al., 2020) mengusulkan model *Generative Emotion Model with Categorized Words* (GEM-CW) untuk klasifikasi sentimen *stock message*. Peneliti menggunakan pendekatan *semi-supervised*. *Generative emotion model* memanfaatkan fitur-fitur pesan, emosi, dan kata secara bersamaan. Hasil eksperimen yang diusulkan menunjukkan model ini efektif, dengan akurasi 74.6%.

2.1.2 Deep Learning (*Approaches*)

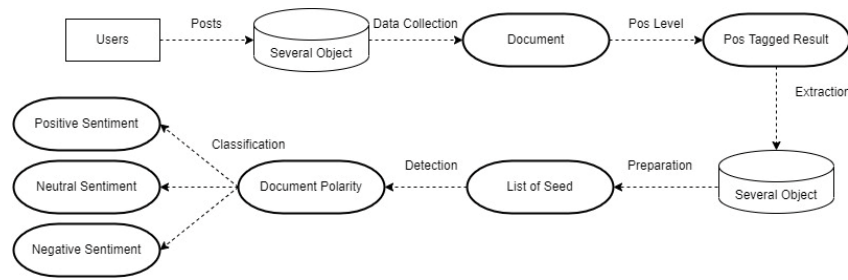
Deep learning adalah sub dari *machine learning* yang menggunakan *artificial neural network* (ANN) / jaringan syaraf tiruan. Belakangan ini *deep learning* telah banyak digunakan untuk analisis sentimen (Ligthart et al., 2021). *Deep learning* diterapkan ke banyak bidang NLP seperti teks, suara, bahasa dan memberikan hasil yang akurat. Klasifikasi teks adalah masalah fundamental di bidang NLP dan media sosial. Penggunaan *deep learning* pada analisis sentimen telah mencapai hasil yang luar biasa (Alshuwaier et al., 2022). Beberapa penelitian terkait analisis sentimen menggunakan pendekatan *deep learning*:

Penelitian yang dilakukan (Majumder et al., 2019) mengusulkan GRU berbasis *neural network* yang bisa dilatih pada *dataset* sentimen dan sarkas. Peneliti menunjukkan bahwa pendekatan berbasis *multitask-learning* memberikan hasil yang baik daripada hanya klasifikasi saja pada *dataset* sentimen dan sarkas. Untuk *dataset* sentimen memberikan presisi 79.89%, *recall* 74.86%, dan *f1-score* 77.30%, dan untuk *dataset* sarkas memberikan hasil presisi 87.41%, *recall* 87.03%, dan *f1-score* 86.97%.

2.1.3 Document-Based (Level of Analysis)

Document-based sentiment analysis (DOCSA) adalah tingkatan analisis sentimen yang sederhana dan menganggap seluruh dokumen sebagai satu komponen (entitas) analisis dan berasal dari satu *stakeholder* (pemegang opini) (Ligthart et al., 2021). Ada tiga tugas pokok dalam DOCSA, yaitu: 1) *extraction opinion words*, 2) *polarity detection*, dan 3) *polarity classification*. *Polarity classification* menentukan dokumen / entitas sebagai positif, negatif atau netral (Alshuwaier et al., 2022). *Polarity classification* diilustrasikan pada gambar 2.2. Beberapa penelitian terkait DOCSA:

Penelitian yang dilakukan (Rhanoui et al., 2019) mengusulkan model CNN-BiLSTM untuk DOCSA. Peneliti menggunakan *dataset* koran nasional dan internasional berbahasa Prancis total 2003 dengan rincian *positive* 474, *neutral* 1247, dan *negative* 282. Untuk *word embedding* menggunakan Doc2vec dan hasil akurasi 90.66%.



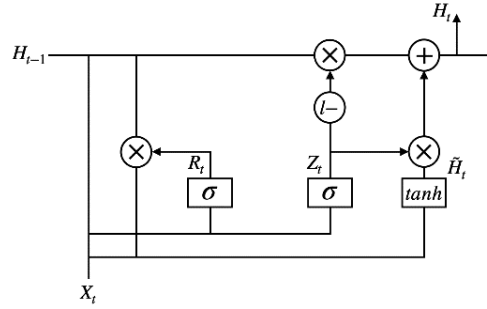
Gambar 2.2 Model polarity classification pada DOCSA

2.2 Bidirectional Gated Recurrent Unit (BiGRU)

GRU adalah salah satu mekanisme dari RNN yang mirip dengan LSTM (Bhuvaneswari et al., 2019). GRU pertama kali diusulkan Gers, dkk. pada tahun 2014 yang merupakan model sederhana dari LSTM. Ada dua *gate* pada GRU yaitu *forget gate* dan *input gate* yang kemudian diteruskan ke *update gate*. Dari *update gate* informasi diteruskan secara selektif ke *hidden layer* untuk mengurangi masalah *gradient* saat mengingat informasi (Zeng et al., 2020). Karena kinerjanya mirip LSTM, GRU cocok digunakan pada penelitian ini dengan karakteristik yang sederhana, parameter yang sedikit, kemampuan menangani *overfitting* yang lebih baik, dan kecepatan konvergensi yang lebih cepat. Untuk lebih jelasnya pada gambar 2.4. Sedangkan Bi-GRU sendiri GRU yang bekerja dari dua arah (Lynn et al., 2019).

Pada penelitian (Han et al., 2020) mengusulkan *Pretraining and Multi-task learning model based on Double BiGRU* (PM-DBiGRU). Pertama peneliti menggunakan bobot *pretraining* dari sentimen klasifikasi ulasan obat untuk Menginisialisasi bobot model usulan peneliti. Kemudian dua Bi-GRU diterapkan untuk menghasilkan representasi semantic dari target dan ulasan obat. Hasil eksperimen menunjukkan bahwa usulan peneliti dapat meningkatkan kinerja

klasifikasi sentimen pada *aspect-based* dibandingkan dengan arsitektur lainnya dengan akurasi 78.26% dan *f1-score* 77.75%.



Gambar 2.3 GRU architecture

Berikut Formula GRU:

Reset gate:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \dots\dots\dots (2.1)$$

Candidate activation vector

$$\tilde{h}_t = \tanh(W_h + [r_t * h_{t-1}, x_t] + b_h) \dots\dots\dots (2.2)$$

Update gate

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \dots\dots\dots (2.3)$$

Candidate activation vector

$$\tilde{h}_t = \tanh(W_h + [z_t * h_{t-1}, x_t] + b_h) \dots\dots\dots (2.4)$$

Hasil hidden gate

$$h_t = (1 - z_t) * h_t + z_t * \tilde{h}_t \dots\dots\dots (2.5)$$

Dimana:

x_t : input vector

h_t : output vector

\tilde{h}_t : candidate activation vector

z_t : update gate vector

r_t : reset gate vector

W, b : parameter matrices and vector

2.3 Attention Mechanism (Transformer)

Attention mechanism adalah mekanisme untuk memfokuskan setiap hasil *output* terhadap target kata. Sehingga kata yang dihasilkan memiliki hubungan yang lebih baik sesuai konteks (Bahdanau et al., 2014). Saat ini, *attention mechanism* telah terbukti efektif untuk mendapatkan hasil yang baik dengan memilih informasi yang penting. *Attention mechanism* pertama kali diusulkan di bidang *computer vision*, tujuannya untuk meniru *attention mechanism* pada manusia dengan memberikan bobot yang berbeda untuk bagian gambar (W. Li et al., 2020). (Bahdanau et al., 2014) menggunakan *attention mechanism* pada mesin penerjemah, dan ini merupakan penerapan *attention mechanism* pertama kali pada bidang NLP. Saat ini, *attention mechanism* telah menunjukkan hasil yang baik dalam bidang NLP, dengan mengimprove *deep neural network* (DNN) dan membiarkan DNN belajar sendiri dimana harus fokus pada konteks (Yadav et al., 2021). Untuk lebih jelas, berikut langkah-langkah *attention mechanism* bekerja:

1. Alignment Score

Alignment Score digunakan untuk mengukur seberapa baik kata *input sequence* dengan *output* pada posisi (t) tersebut. Caranya adalah dengan mengambil *encoded* pada *hidden state* (h_i) dan *decoder output* sebelumnya (S_{t-1}) untuk menghitung nilai $e_{t,i}$. *Alignment model* disimbolkan dengan fungsi $a(.)$. Berikut persamaan dari *alignment score*.

$$e_{t,i} = a(s_{t-1}, h_i) \dots\dots\dots (2.6)$$

2. Attention Weight

Setelah ketemu nilai (S_{t-1}) pada perhitungan *alignment score*. Kemudian menghitung bobot/weight $(a_{t,i})$. yaitu dengan memasukkan *alignment score* ke dalam fungsi *softmax*. Fungsi aktivasi *softmax* akan menghasilkan probabilitas yang digunakan untuk mewakili bobot pada setiap *input sequence*, semakin tinggi bobot maka semakin tinggi pula pengaruh atau kontribusi yang diberikan. Berikut persamaan dari *attention weight*.

$$a_{t,i} = \text{softmax}(e_{t,i}) \dots\dots\dots (2.7)$$

3. Context Vector

Langkah terakhir adalah *context vector*. *Context vector* berfungsi untuk menghitung *output* akhir dari *decoder*. *Context vector* adalah jumlah dari *weight* $(a_{t,i})$ dan *hidden state decoder* (h_i) . Berikut persamaan dari *context vector*.

$$c_t = \sum_{i=1}^T a_{t,i} h_i \dots\dots\dots (2.8)$$

Dari persamaan ini bisa dilihat bahwa $(a_{t,i})$ mengisyaratkan seberapa banyak (h_i) berkontribusi pada konteks.

Penelitian (Zhu et al., 2019) mengusulkan model LSTM dua arah untuk meningkatkan *attention*. *Attention mechanism* digunakan secara langsung untuk mempelajari bobot setiap kata pada sentimen kalimat berdasarkan *word vector*. LSTM digunakan untuk mempelajari informasi semantic teks, dan meningkatkan klasifikasi melalui *parallel fusion*. penelitian (Ma et al., 2018) mengusulkan metode baru untuk *aspect-based*. Metode ini menggunakan *layer attention mechanism* yang terdiri dari *target-level attention* dan *sentence-level attention* untuk menambah jaringan LSTM. Eksperimen dilakukan pada dua dataset publik,

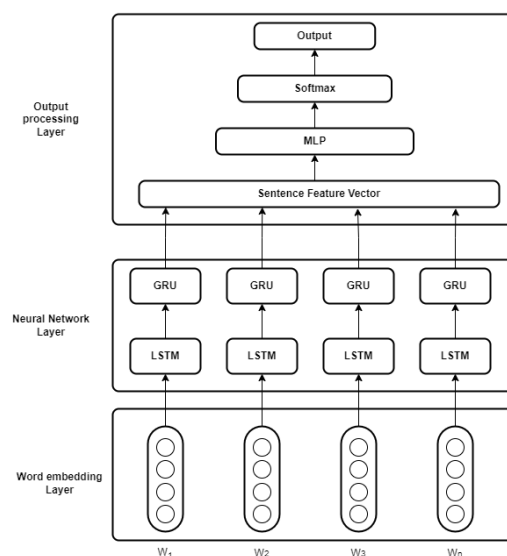
dan hasilnya menunjukkan bahwa arsitektur *attention* yang diusulkan dikombinasikan dengan *Sentic LSTM* dapat mengungguli metode terbaru dalam *task* klasifikasi emosi.

2.4 Novelty Penelitian

Sudah banyak penelitian terkait analisis sentimen yang telah diusulkan. Beberapa penelitian tentu memiliki ciri khas dan karakteristik tersendiri. Ciri khas dan karakteristik penelitian pada analisis sentimen itu sendiri ada pada *dataset*, *word embedding layer*, *neural network layer* dan *classifier*. Beberapa usulan arsitektur metode analisis sentimen sebagai berikut:

1. Arsitektur LSTM-GRU + GloVe

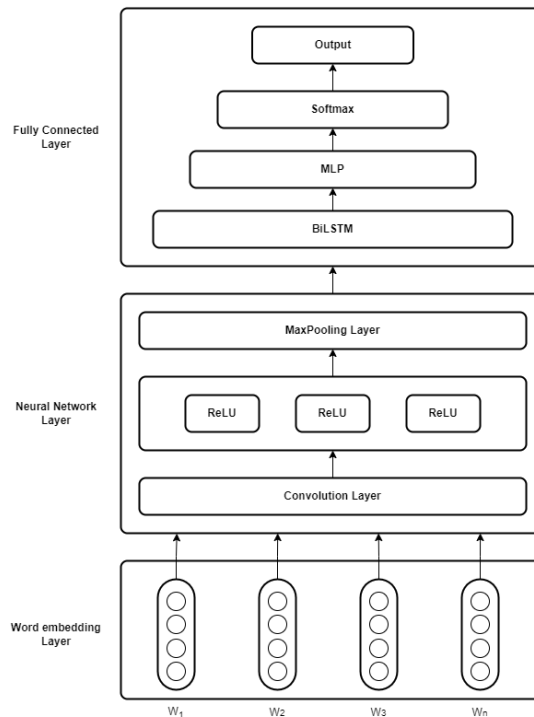
Arsitektur ini diusulkan oleh (Ni & Cao, 2020). Pada *word embedding layer* menggunakan *GloVe*, *neural network layer* menggunakan gabungan LSTM dan GRU, dan untuk *classifier* menggunakan *softmax*. Berikut arsitektur LSTM-GRU + GloVe:



Gambar 2.4 Arsitektur LSTM-GRU + GloVe

2. Arsitektur CNN-BiLSTM + Doc2vec

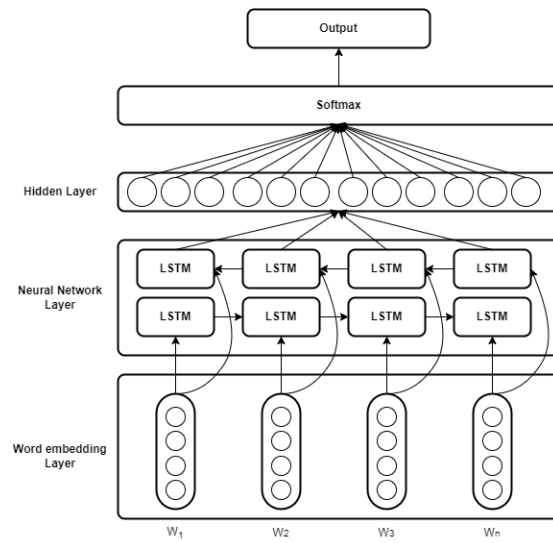
Arsitektur ini diusulkan oleh (Rhanoui et al., 2019). Pada *word embedding layer* menggunakan *Doc2Vec*, *neural network layer* menggunakan gabungan CNN dan BiLSTM, dan *classifier* menggunakan *softmax*. Berikut arsitektur CNN-BiLSTM + Doc2vec:



Gambar 2.5 Arsitektur CNN-BiLSTM + Doc2Vec

3. Arsitektur BiLSTM + Seninfo-TFIDF

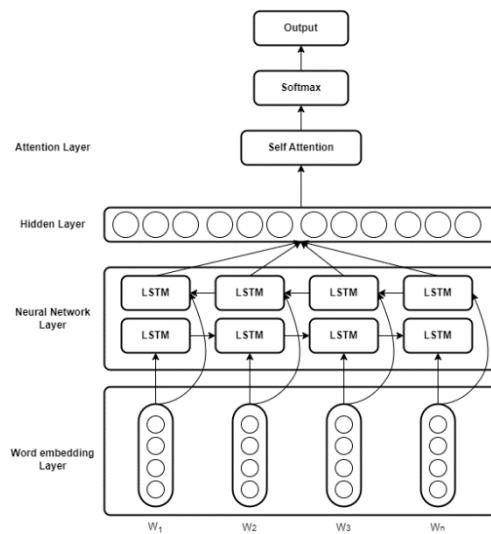
Arsitektur ini diusulkan oleh (Xu et al., 2019). Pada *word embedding layer* menggunakan gabungan *sentiment information* (Seninfo) dan TFIDF, *neural network layer* menggunakan Bi-LSTM, dan *classifier* menggunakan *softmax*. Berikut arsitektur BiLSTM+Seninfo-TFIDF:



Gambar 2.6 Arsitektur BiLSTM+seninfo-TFIDF

4. Arsitektur SAM-BiLSTM + GloVe

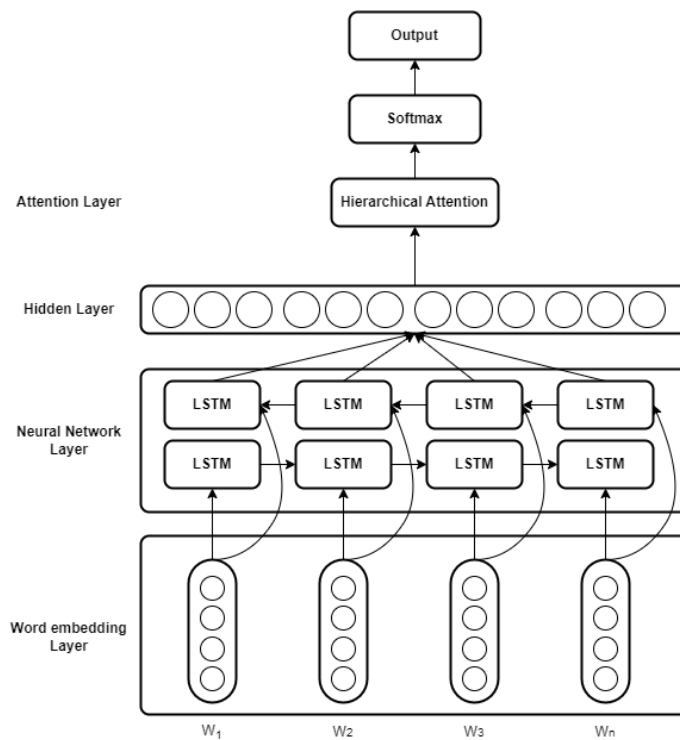
Arsitektur ini diusulkan oleh (W. Li et al., 2020). Pada *word embedding layer* menggunakan GloVe, *neural network layer* menggunakan Bi-LSTM, *attention layer* menggunakan *self-attention*, dan *classifier* menggunakan *softmax*. Berikut arsitektur SAM-BiLSTM:



Gambar 2.7 Arsitektur SAM-BiLSTM

5. Arsitektur BiLSTM-HAN

Arsitektur ini diusulkan oleh (J. Li et al., 2019). Pada *word embedding layer* menggunakan *one-hot encoding*, *neural network layer* menggunakan BiLSTM, *attention layer* menggunakan *hierarchical attention*, dan *classifier* menggunakan *softmax*. Berikut arsitektur BiLSTM-HAN:

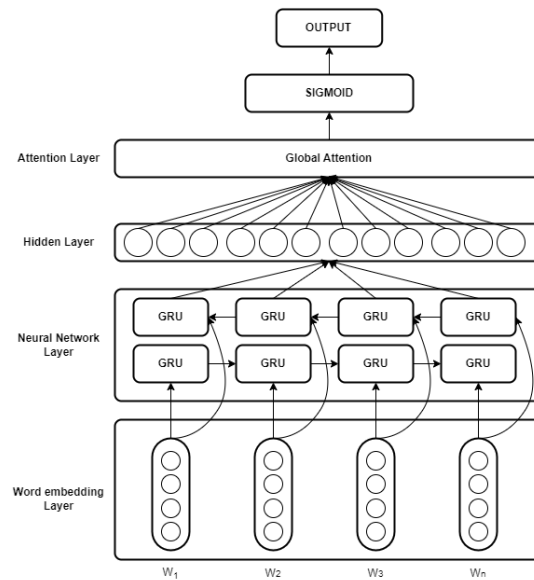


Gambar 2.8 Arsitektur BiLSTM-HAN

6. Arsitektur Glo-Att-BiGRU+Word2vec

Arsitektur ini merupakan arsitektur analisis sentimen yang diusulkan. Pada *word embedding layer* menggunakan *word2vec*, *neural network layer* menggunakan BiGRU, *attention layer* menggunakan *global*

attention, dan *classifier* menggunakan *sigmoid*. Berikut arsitektur Glo-Att-BiGRU+Word2vec:



Gambar 2.9 Arsitektur Glo-Att-BiGRU+Word2vec

Pemaparan masing-masing arsitektur analisis sentimen yang diusulkan sebelumnya (J. Li et al., 2019; W. Li et al., 2020; Ni & Cao, 2020; Rhanoui et al., 2019; Xu et al., 2019) bertujuan dalam rangka menyusun *novelty* penelitian. Adapun *novelty* pada penelitian ini terdapat pada tabel 2.1:

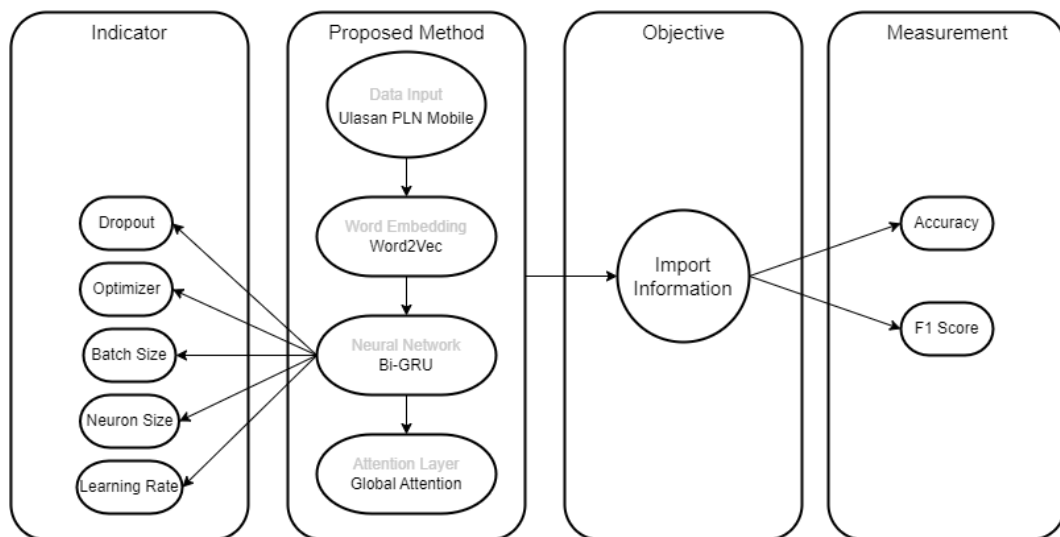
Tabel 2.1 Novelty penelitian

Arsitektur	Word embedding	Neural network	Attention	Classifier
(Ni & Cao, 2020)	GloVe	LSTM+GRU	-	Softmax
(Rhanoui et al., 2019)	Doc2Vec	CNN+BiLSTM	-	Softmax
(Xu et al., 2019)	Seminfo+TFIDF	BiLSTM	-	Softmax
(W. Li et al., 2020)	GloVe	BiLSTM	Self-attention	Softmax
(J. Li et al., 2019)	One-hot encoding	BiLSTM	Hierarchical-attention	Softmax
Arsitektur yang diusulkan	Word2vec	BiGRU	Global-attention	Sigmoid

Berdasarkan matriks penelitian pada tabel 2.1, maka *novelty* penelitian ini ada empat, yaitu ada pada: *word embedding layer*, *neural network layer*, *attention layer*, dan *classifier*.

2.5 Kerangka Teori

Kerangka teori adalah suatu diagram yang menjelaskan garis besar dari alur sebuah penelitian. Kerangka teori dirancang berdasarkan rumusan masalah (*research question*), dan merepresentasikan suatu himpunan dari beberapa konsep serta hubungan antara konsep-konsep tersebut. Kerangka teori pada penelitian ini terdapat pada gambar 2.4



Gambar 2.10 Kerangka teori

Pada gambar 2.4 komponen kerangka teori dibagi menjadi empat, berikut penjelasan dari masing-masing komponen:

1. *Indicator*

Indicator (Indikator) adalah variabel yang dilakukan untuk observasi.

Penelitian ini menggunakan *hyperparameter setting* pada Bi-GRU untuk tujuan observasi.

2. *Proposed Method*

Proposed method (model yang diusulkan) merupakan kombinasi dari BiGRU untuk *sequence learning* dan *Global Attention* untuk memfokuskan kata yang paling besar memberikan kontribusi. Dengan *Word embedding* menggunakan *word2vec*.

3. *Objective*

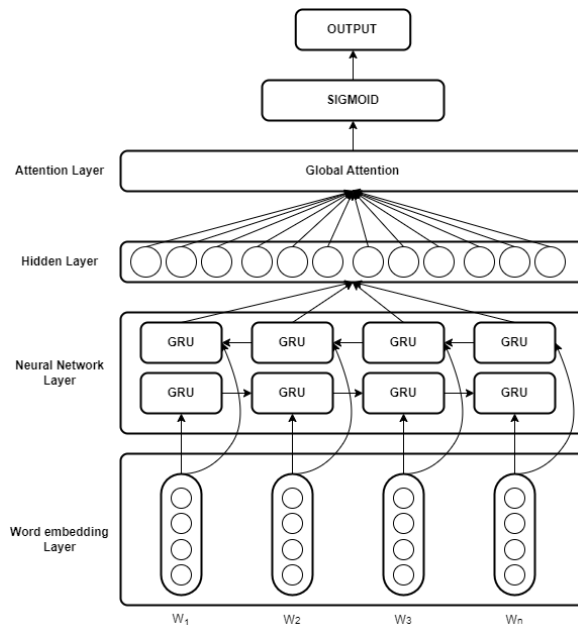
Tujuan (*objective*) pada penelitian adalah untuk menangkap informasi penting dari *context information* yang dihasilkan Bi-GRU.

4. *Measurement*

Pengukuran (*measurement*) menggunakan dua tolak ukur yaitu akurasi dan F1 score sebagai acuan dari keberhasilan menyelesaikan masalah.

3.4 Model yang Diusulkan (*Proposed Method*)

Pada bagian ini, penulis memaparkan usulan metode secara rinci. Pada penelitian ini penulis mengusulkan *attention mechanism* yang digabungkan dengan Bi-GRU. Arsitektur *proposed method* ditunjukkan pada gambar 3.7.



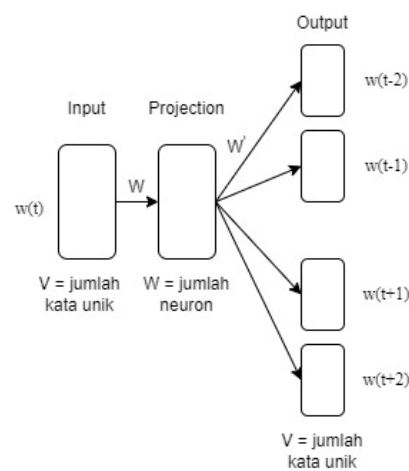
Gambar 3.8 Arsitektur model Bi-GRU dengan attention mechanism

Penulis mengusulkan penelitian analisis sentimen ulasan PLN Mobile terdiri dari: 1) *Task*: sentimen klasifikasi, 2) *Approach*: *deep learning*, 3) *Level- of Analysis*: *Document Based*. Dan terbagi tiga layer. Pertama, dilakukan *word embedding* sebelum *training* untuk merepresentasikan *word*. Kedua, proses *sequence learning* menggunakan Bi-GRU. Ketiga, *attention layer* untuk menangkap informasi penting dari Bi-GRU.

1. *Word Embedding*

Tradisional *text representation*, seperti *Bag of Words*, terdapat dua masalah, yaitu: kehilangan *learning representation* dan ukuran

dimensionality yang besar. Untuk mengatasi masalah ini, penelitian ini menggunakan *word embedding*, untuk merepresentasikan setiap kata yang mirip secara semantik. *Word embedding* yang digunakan adalah *word2vec* yang diusulkan oleh Mikolve (Mikolov et al., 2013). Arsitektur yang digunakan dari *word2vec* adalah skip-gram. Model dilatih dengan memaksimalkan *average log probability* di semua kata.



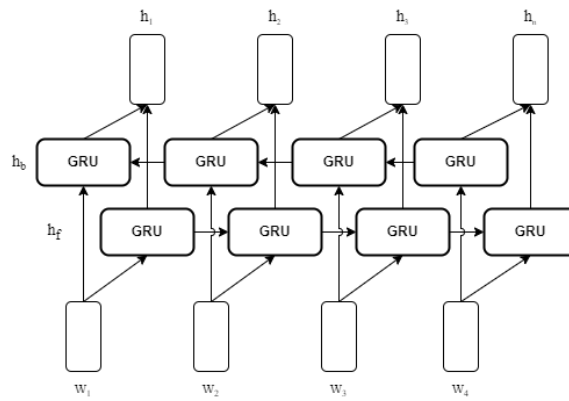
Gambar 3.9 Arsitektur skip-gram

Pada gambar 3.8 arsitektur skip-gram memberikan bobot pada W dan W' secara acak. W dan W' merupakan matrik dengan ukuran $W = V * N$ dan $W' = N * V$. Pada tahap *feedforward*, *input* di *dot product* dengan W dan menghasilkan nilai pada *hidden layer (projection)*. Kemudian *hidden layer* di *dot product* dengan W' dan menghasilkan nilai *output*, setelah itu akan dihitung nilai errornya menggunakan *cross entropy*. Pada tahap *backpropagation*, menggunakan teknik *gradient descent* yaitu dengan melakukan *upgrade* W dan W' . Proses iterasi akan terus berjalan sampai

nilai error minimum. Penelitian ini menggunakan ukuran dimensi setiap *word vector* sebesar 300.

2. Bi-GRU

Secara konsep, analisis sentimen adalah proses *sequence learning* dengan jenis *task many-to-one*. *Word representation* yang dihasilkan oleh *word2vec* tidak terdapat *sequence information*. Karena itu, Bi-GRU digunakan untuk *sequence modelling* yang dapat mengekstrak *context information* dari *word representation* yang dihasilkan *word2vec*.



Gambar 3.10 Arsitektur bidirectional GRU

Pada gambar 3.9 Bi-GRU mempelajari *context information* dari dua arah yaitu: *forward* dan *backward*, setelah itu menggabungkan keduanya. Bi-GRU terdiri dari *forward* GRU (direpresentasikan sebagai \overrightarrow{GRU}) untuk mempelajari *feature sequence* dari Lc_1 ke Lc_{100} dan *backward* GRU (direpresentasikan sebagai \overleftarrow{GRU}) untuk mempelajari *feature sequence* dari Lc_{100} ke Lc_1 . Berikut formula Bi-GRU secara detail:

$$\vec{h}_f = \overrightarrow{GRU}(Lc_n), n \in [1, 100] \dots\dots\dots (3.1)$$

$$\overleftarrow{h}_b = \overleftarrow{GRU}(Lc_n), n \in [100, 1] \dots\dots\dots (3.2)$$

3. *Global Attention*

Pada dokumen sentimen, setiap kata memiliki kontribusi yang berbeda terhadap konteks sentimen, dengan memberikan bobot yang berbeda pada kata, dapat memberikan fokus (*attend*) yang berbeda pada *context information* yang dihasilkan Bi-GRU. *Global attention* digunakan untuk memberikan bobot yang berbeda pada kata untuk meningkatkan pemahaman terhadap sentimen. *Global attention* dapat memfokuskan fitur dari *keyword* dan mengurangi fitur *non-keywords* menggunakan *fully-connected layer* dan *softmax function*. Berikut formula dari *global attention*:

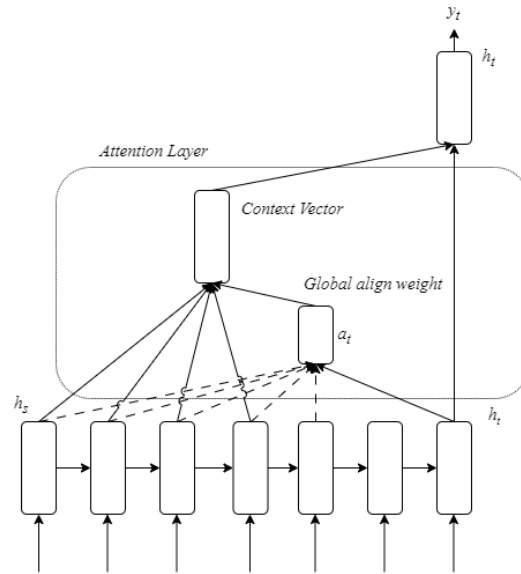
$$M = \tanh \left(\begin{bmatrix} W_h H \\ W_v v_a \otimes e_n \end{bmatrix} \right) \dots\dots (3.3)$$

$$a = \text{softmax}(W^T M) \dots\dots (3.4)$$

$$r = H a^T \dots\dots (3.5)$$

Dimana $H \in \mathbb{R}^{d \times N}$ adalah matriks yang terdiri dari *hidden vector* $[h_1, \dots, h_N]$ yang dihasilkan Bi-GRU, d adalah ukuran dari *hidden layer*. v_a adalah representasi *embedding* dan W adalah yang harus dipelajari selama *training*.

Pada formula 3.3 digunakan untuk mencari *alignment score* menggunakan *hyperbolic tangent* dan menghasilkan fitur yang lebih kompleks. Formula 3.4 digunakan untuk mencari *attention weight* dengan menambahkan *softmax* dan menghasilkan distribusi sentimen. Formula 3.5 digunakan untuk mencari *context vector*.



Gambar 3.11 Arsitektur global attention

Pada gambar 3.10 setiap *time step* t , *global attention* menghitung *alignment weight* a_t berdasarkan *target state* saat ini h_t . *context vector* c_t dihitung sebagai bobot rata-rata berdasarkan *alignment weight* dan semua *state*.

3.5 Eksperimen dan Pengujian Model (*Model Test and Experiment*)

Eksperimen dilakukan untuk menguji model yang diusulkan. Penelitian ini menggunakan empat skenario pengujian yang berbeda untuk mencari akurasi model yang terbaik. Skenarionya adalah dengan melakukan *hyperparameter setting*. Berikut skenario-skenario pengujian:

1. Melakukan *tuning* pada *dropout*

Skenario pertama dengan melakukan *tuning* pada *dropout*. Yaitu dengan mengurangi *unit neural network* pada ukuran tertentu. ukuran tersebut seperti yang tertera pada tabel 3.2: